UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of a master's thesis by

Penghuan Ni

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

_____

Name of Faculty Advisor

_____

Signature of Faculty Advisor

_____

Date

GRADUATE SCHOOL

Your Great Title For Your Thesis

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Penghuan Ni

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Professor Advisor Name

December 2015

Acknowledgements

Some acks are nice.

Dedication

Your dedication goes here.

Abstract

We are trying to apply Nondominated Sorting Genetic Algorithm II(NSGA-II) to find the Pareto Optimal Solutions of our Urban Optimization Problem. There are different types of Multi-Objective Genetic Algorithms(MOGA), based on the previous research [1][6], we are interested to see how NSGA-II works on our problem. To solve this problem, we implement and modify NSGA-II to fit our problem well. We expect that NSGA-II could give us multiple Pareto Optimal Solutions and help us to make a better decision.

Contents

List of Figures

1 Introduction

It's always good to introduce your (1) problem, (2) why it is interesting, (3) what you did, and (4) roughly, how well did it work. You might even have citations in here, as in this paper [**Asawa:2008:TDT**].

2   Background

2.1   Background

Based on the previous work of people using NSGA-II on some multi-objective optimization problems[6], we believe we can also applied NSGA-II on our Urban Design Model to find the optimal solution.

2.2   Previous Work

2.2.1   Genetic Algorithm(GA)

After the Genetic Algorithm was introduced in 1975 by John Henry Holland[5], it has been applied in many areas. With mimicking the process of natural selection, GA uses selection, crossover, and mutation to generate solutions and tries to find the optimal solution for our problems. More specifically, we have an initial population size, then based on our criterion or objective, we give each individual a fitness value. After we select a certain number of individuals from our population based on their fitness, then we apply our crossover function with a crossover probability to have some new offsprings. Next, we apply the mutation function with a mutation probability on the new offsprings. After all those steps, hopefully we can have a new population which have some better fitness values. Last, we repeat our steps until we achieve our optimal solution.

2.2.2 Multiobjective Genetic Algorithm(MOGA)

When we apply GA to find our optimal solutions, we can divide those problems into two categories based on the number of objectives: Single Objective problems and Multi-Objective problems. Let's take a simple example for single objectives first. For example we want to find the minimum value of $y = x^2$ with $x \in [-3, 3]$, it is easy to see that we could achieve the minimun value at $x = 0$. When we have two objectives, the problem becomes more complicated. For example we want to minimize both $y_1 = x^2$ and $y_2 = (x-2)^2$. Then, we are not able to find a specific x value for this problem. From Figure 2.1, we could see that x at 0 could give us minimum value of $y_1 = x^2$ and x at 2 could give us minimum value of $y_2 = (x - 2)^2$. However we are not able to find a x value which could guarantee us both y_1 and y_2 at their minimun value. We call $x = 0$ or 2 Pareto Optimal Solutions[4][9]. You could find more information about Pareto Optimal Solutions at our latter sections. Other than $x = 0$ or 2, any points between 0 and 2 could be a valid compromise solution.

Figure 2.1: Functions $y_1 = x^2$ and $y_2 = (x - 2)^2$

In the real word, there are many kinds of simultaneous optimization of multiple objectives problems. People have tried many methods to find Pareto Optimal Solution. The classical method is to convert multiobjective problem to a single objective problem. In paper [8], three classical methods are introduced: 1. Objective Weighting, 2. Distance

Functions, and 3. Min-Max Formulation. All those methods have one common drawback: it can only give us a single-point solution. However, when decision makers make decision, they often need different alternatives. Another significant drawback of those method is they heavily depend on what weight vector or demand level we choose, which demands that the user have knowledge about the underlying problem. However, in most cases, users do not have much knowledge about the problem and that is one of the reasons they use genetic algorithm. So, those methods are not efficiency and we probably need to try many times to get a acceptable solution.

Since classical methods to solve multiobjective optimization problems are inadequate and inconvenient, people start to find other ways to implement GA. Next, we will briefly introduce some of those implementations.

2.2.3 Vector Evaluated Genetic Algorithm(VEGA)

VEGA is the first practical algorithm and was developed by Schaffer in 1984[7]. Instead of changing multiple objectives problem to a single objective problem, Schaffer modified the simple tripartite genetic algorithm by performing independent selection cycles according to each objective[7]. Bascially, to fill up a portion of the mating pool he created a loop around the traditional selection procedure to repeated selection method for each individual objective. After this selection step, we could have a thoroughly shuffled population to apply crossover and mutation steps. This could make the mating of individuals from different subpopulation groups possible.

With the independent selection of specialists, we could make the population in speciation. After a large number of generations, we could see the convergence of the entire population toward the individual optimum regions. To make a decision, we may not want to have any bias on such middling individuals, rather, we desire more nondominated points.Schaffer

developed two heuristics to minimize this speciation: the nondominated selection heuristic and the mate selection heuristic. For the nondominated selection heuristic, he penalized the dominated individuals by substracting a small fixed penalty from their expected number of copies druing selection. Then the total penalty for dominated individuals was divided among the nondominated individuals and was added to their expected number of copies during selection. For the mate selection heuristic, he wanted to promoted the cross-breeding of specialists from different subgroups by selecting an individual, as a mate to a randomly selected individual, which has the maximum Euclidean distance in the performance space from its mate. However, both of those two heuristics are failed due to some certain problems.

2.2.4 Nondominated Sorting Genetic Algorithm(NSGA)

NSGA was first introduced by N. Srinivas and Kalyanmoy Deb in 1994[8]. They investigated Goldberg's notion of nondominated sorting in GAs along with a niche and speciation method to find multiple Pareto Optimal Solutions simultanesously. The reason they call this algorithm the Nondominated Sorting Genetic Algorithm is that it build on a nondominated sorting procedure. Two ideas behind the nondominated sorting procedure are ranking selection which is used to emphasize good points method and niche method which is used to maintain stable subpopulations of good points.

The only difference between simple genetic algorithm and NSGA is the way selection operator works. First, we rank the population based on the individuals' nondomination and identify those nondominated individuals from the current population. Second, let all those individuals constitute the first nondominated front and assign a large dummy fitness value to them. All those nondominated individuals which have an equal reproductive potential should have same fitness value. Third, to maintain diversity in the population, we apply

sharing methods[3][2] on these classified individuals. Sharing is achieved by performing selection operation using degraded fitness values that are obtained by dividing the original fitness value of an individual by a quantity proportional to the number of individuals around it. Last, ignore those nondominated multiple optimal points and using the same method to process the rest of population to get the second nondominated front. This time, we assign a new dummy fitness value to the second nondominated front which is smaller than the first nondominated front. By this process, we could classified the entire population into several fronts. After the classification process, we reproduce the population by each individual's dummy fitness value and apply a stochastic remainder proportionate seletion on them.

After this selection process, we move to crossover operation. Since we want to find nondominated regions or Pareto Optimal fronts and first front have the maximum fitness value, we give them more copies. This could give us a fast convergence of the population toward nondominated regions. Furthermore, sharing method could help us to distribute the population over this region. By nondominated sorting procedure, NSGA becomes more efficient. And NSGA can solve both minimization and maximization problems. But it is still not very handy.

2.2.5 Nondominated Sorting Genetic Algorithm II(NSGA-II)

NSGA-II[1] is a very famous multi-objective optimization algorithm which is wide used to find the optimal solution nowadays. Compare with the previous NSGA, NSGA-II has improved the computational efficiency, elitism, and sharing parameter. For computational complexity of nondominated sorting, it improve from $O(MN^3)$ to $O(MN^2)$. With the elitism, NSGA-II could not only speed up the GA performance, but also prevent loss of good solutions. Moreover, NSGA II does not need to specify a sharing parameter σ_{share}, which is a requirement for pervious multi-objective evolutionary algorithms.

2.2.6　Pareto Optimal Solution

3   Implementation

3.1   First section

You may need a nice figure, which you can algorithmically render using the Tikz package. You should really check out the Texample web site where several nice tikz examples are provided (http://www.texample.net/tikz/examples/all/).

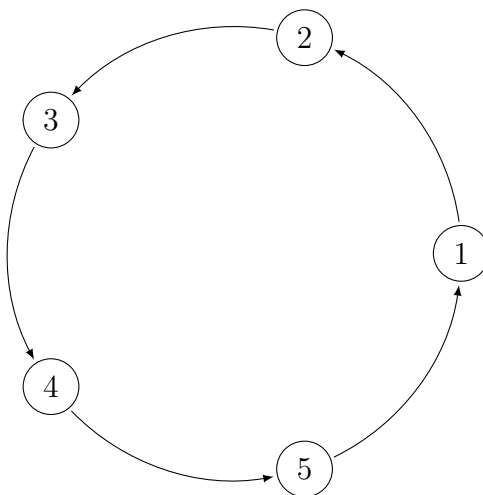A graph rendered with the Tikz package is shown in Figure 3.1.

Figure 3.1: Clear and concise figure captions are important to write. This one illustrates the cycle of a graph.

3.1.1 Subsection One

3.1.2 Subsection Two

3.1.3 Subsection Three

3.2 New Section For Next Important Topic

3.2.1 Algorithm Initialization

3.2.2 Atomic Operations

You may even need code in your thesis. Here is a way to nicely include code with LATEXusing the listings package.

```
1  for (unsigned int idx=0; idx<maxSize; idx++) {
2     atomic_add( idx );
3  }
```

3.2.3 Programming Style

Explaining Fine Detail Here

TODO: Make sure to finish this!

Last Subsection

4   Results

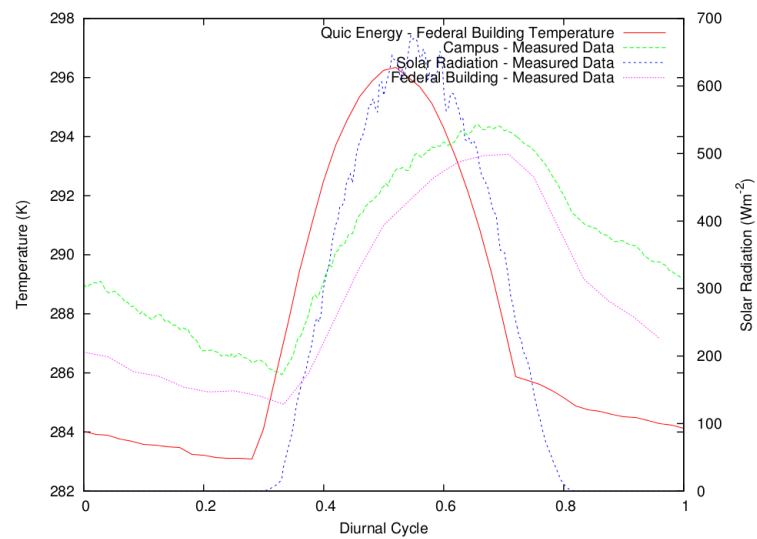Your results. This worked great. Here's a plot to show how great it worked.

Figure 4.1: Good data.

Figure 4.1

5 Conclusions

How can you wrap this up?

A   Appendix A

Do you need an Appendix? You can include several of them if you want.

Bibliography

[1]   K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. ``A fast and elitist multiobjective genetic algorithm: NSGA-II''. In: *Evolutionary Computation, IEEE Transactions on* 6.2 (Apr. 2002), pp. 182–197. ISSN: 1089-778X. DOI: 10.1109/4235.996017 (cit. on pp. iii, 6).

[2]   K. Deb. *Genetic algorithms in multimodal function optimization*. Clearinghouse for Genetic Algorithms, Department of Engineering Mechanics, University of Alabama, 1989 (cit. on p. 6).

[3]   K. Deb and D. E. Goldberg. ``An investigation of niche and species formation in genetic function optimization''. In: *Proceedings of the 3rd international conference on genetic algorithms*. Morgan Kaufmann Publishers Inc. 1989, pp. 42–50 (cit. on p. 6).

[4]   A. Hans. ``Multicriteria optimization for highly accurate systems''. In: *Multicriteria Optimization in Engineering and Sciences* 19 (1988), pp. 309–352 (cit. on p. 3).

[5]   J. Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. University of Michigan Press, 1975. ISBN: 9780472084609. URL: https://books.google.com/books?id=YE5RAAAAMAAJ (cit. on p. 2).

[6]   L. Magnier and F. Haghighat. ``Multiobjective optimization of building design using TRNSYS simulations, genetic algorithm, and Artificial Neural Network''. In: *Building and Environment* 45.3 (2010), pp. 739–746 (cit. on pp. iii, 2).

[7]   J. D. Schaffer. ``Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms (Artificial Intelligence, Optimization, Adaptation, Pattern Recognition)''. AAI8522492. PhD thesis. Nashville, TN, USA, 1984 (cit. on p. 4).

[8]   N. Srinivas and K. Deb. ``Muiltiobjective optimization using nondominated sorting in genetic algorithms''. In: *Evolutionary computation* 2.3 (1994), pp. 221–248 (cit. on pp. 3, 5).

[9]   C. Vira and Y. Y. Haimes. *Multiobjective decision making: theory and methodology*. 8. North-Holland, 1983 (cit. on p. 3).